Continuous head pose estimation using manifold subspace embedding and multivariate regression

Continuous head pose estimation using manifold subspace embedding and multivariate regression

Katerine Diaz-Chito, Jesús Martínez del Rincón, Aura Hernández-Sabaté, Debora Gil,*Serra Hunter Fellow*

***Abstract*—In this paper, a continuous head pose estimation system is proposed to estimate yaw and pitch head angles from raw facial images. Our approach is based on manifold learning-based methods, due to their promising generalization properties shown for face modelling from images. The method combines histograms of oriented gradients, generalized discriminative common vectors and continuous local regression to achieve successful performance. Our proposal was tested on multiple standard face datasets, as well as in a realistic scenario. Results show a considerable performance improvement and a higher consistence of our model in comparison with other state-of-art methods, with angular errors varying between 9 and 17 degrees.**

***Index Terms*—Head pose estimation, HOG features, Generalized Discriminative Common Vectors, B-splines, Multiple linear regression.**

I. Introduction

Accurate head pose estimation is a challenging problem in itself due to the variability introduced by multiple factors such as illumination, identity and expression, to name a few. During the last decade there has been an increasing interest in developing head pose estimation methods for different applications such as security and surveillance systems [1], human-robot interaction [2], meeting rooms [3], intelligent wheelchair systems [4], and driving monitoring [5], [6], [7], [8]. Head pose is typically expressed by three angles (yaw, pitch and roll) that describe the orientation with respect to a head-centered frame, being yaw and pitch the angles that are more related with the gaze and attention of the subject under consideration.

Automatic head pose estimation has been approached from different points of view, from appearance based methods, such as manifold embedding, regression or classification approaches, to model based methods, which includes deformable and geometric models. While model-based methods exhibit excellent performance, specially in frontal images or small angles, they require detecting/tracking facial features with high precision and they are significantly affected by partial occlusions of such facial landmarks or by illumination changes, common in real environments. On the contrary, appearance based methods are less sensitive to partial occlusions and extreme angular views since these approaches use the full image of the head, but at the cost of higher computational cost.

In this paper, we propose a novel appearance based approach for both yaw and pitch estimation that combines an advance manifold embedding with regression in order to achieve state-of-art performance. Furthermore, Histogram of Oriented Gradients (HOG) are extracted from the image as preliminary feature extraction step. Our approach can be applied to both full head images or to face crops in combination with a face detector. Our system is thoroughly evaluated in 6 different datasets and compared against state-of-art methods HPE [9], [10], DRMF [11] and OPENFACE [12]. The main contributions of our paper are the proposal of a manifold embedding based on discriminative common vectors that allows a better modelling of the face image subspace, and a fully continuous regression model that allows continuous angle estimation, including extreme angles.

This paper is structured as follows: Section I-A briefly introduces the related works in this field. Section II introduces the method proposed. Section III describes the empirical validation and presents the results and the analysis of the proposed approach as well as its comparison against the state of the art. Finally, Section IV summarizes the main conclusions and results.

A. Related work

This section is limited to the most relevant literature to our work, the appearance-based methods, i.e. those methods that use the full raw image as input due to their advantages for real unconstrained environments. A complete description of all the methods available is out of the scope of this article so we refer the reader to the survey [13], the paper [14] and the book [15], although theses do not include methods based on depth learning due to their recent appearance.

Appearance based methods have historically considered the head estimation problem as a discrete problem -i.e. as a classification problem-, or as a continuous problem -i.e as a regression problem. Classification-based methods [13], [14], [15] suffer from granularity of the estimated angles given the difficulty to train two classes whose angles are very close. In contrast, regression based methods provide a fully continuous estimation, resulting on a higher proliferation of these approaches in the literature. These approaches are mainly composed of two main stages: a first stage where a feature set is obtained from the raw image, and a second stage where linear/nonlinear regression methods make use of a labelled training set to create a mapping from images/features space to their corresponding poses.

The feature extraction techniques employed vary from applying conventional HOG features [16], to automatic feature extraction using linear manifold embedding, such as principal component analysis, or using complex non linear methods such as convolutional neural networks (CNN) architectures. All these techniques aim to create a discriminative feature space

This work was supported by the Spanish projects DPI2015-65286-R, 2017 SGR 1624 and the CERCA Programme.

where the correspondence between the feature space location and the pose is easy to establish. Thus, Huang et al. [17] used supervised local subspace learning to learn a local linear model which showed prominent potential to provide accurate head pose estimation when the training data is pretty sparse and non-uniformly sampled. Haj et al. [18] applied partial least squares regression to model the relationship between observed variables by projecting them into a latent space. This alleviates the negative effect on pose estimation when there exists misalignment of head location in the image. Wang et al. [19], [20] presented a framework under the neighborhood construction, graph weight computation and projection learning. They redefined inter-point distance for neighborhood construction as well as graph weight by constraining them with the pose angle information. Then, they used a supervised neighborhood-based linear feature transformation algorithm to keep the data points with similar pose angles close together but the data points with dissimilar pose angles far apart. Peng et al. [21] proposed a coarse-to-fine pose estimation framework in the latent space, where the unit circle and 3-spheres are employed to model the manifold topology on the coarse and fine layers respectively. Chen et al. [16] estimated the head pose by using gradient-based features and support vector regression to low resolution images. Recently, Drouard et al. [9], [10] proposed to use a mixture of linear regressors with partially-latent output. First, the bounding box containing the face is re-sized to 64×64, converted to a grey-level image to which histogram equalization is then applied. A HOG descriptor is extracted from this patch, such as a HOG pyramid is build by stacking HOG descriptors at multiple resolutions. Then, the proposed regression method learns a map from this high-dimensional feature vector onto the joint space of head-pose angles and bounding-box shifts.

Recent advances in deep learning made possible to easily train complex neural networks on large datasets, leading to staggering progress in many different fields from natural language processing to image processing, due to their ability to automatically derived discriminative features, as it is the case of CNNs. This has also been applied to the head pose estimation problem, where many approaches have been proposed. Foytik et al. [22] presented a pose estimation framework that seeks to describe the global nonlinear relationship in terms of localized linear functions. A two layer system is formulated on the assumptions that coarse pose estimation can be performed adequately using supervised linear methods, and fine pose estimation can be achieved using linear regressive functions if the scope of the pose manifold is limited. Ahn et al. [23] proposed a head pose estimation algorithm for monocular camera, by using a convolutional filters and exploiting the neural architecture in a data regression manner to learn the mapping function between visual appearance and three dimensional head orientation angles. Patacchiola et al. [24] also proposed an approach based on CNNs based on a divide-and-conquer strategy, training different CNNs for each degree of freedom. However, while CNN based methods provide excellent performance if huge amount of training data is available, this performance is only exhibited in the same type of images and conditions present in training , due to severe overfitting to the training set [25]. Their performance in cross-dataset testing or realistic scenarios with little available data decreases rapidly, in contrast to the manifold embedding techniques which shows promising generalisation properties [15].

II. Head pose estimation

Our head pose estimation framework is composed of three main components: an initial feature extraction based on the computation of Histogram of Oriented Gradients, a manifold embedding projection based on Generalized Discriminative Common Vectors (GDCV), and a continuous regression composed of spline fitting and multivariate local regression. This pipeline, as well as the resulting subspaces involved at each step, are depicted in Figure 1.

A. HOG feature extraction

First, HOG features are extracted to enhance the discriminative information in the image [9], [10] before the final feature embedding space is calculated. The underlying idea is that local object appearance and shape are well characterized by the distribution of local intensity gradients and edge directions while being less sensitive to illumination changes and cluttered and changing background. While the manifold embedding could be generated directly using the raw image as direct input (see Section III-C), the resulting space using HOG feature enhances the discrimination between pose orientations.

Locally normalized Histogram of Oriented Gradient (HOG) descriptors [26] were selected as in [9], [10] due to their excellent performance to detect edge orientation, relative to other existing feature sets. The implementation of these HOG descriptors can be achieved by dividing the image into small connected regions (cells), and for each cell computing a histogram of gradient directions (i.e. edge orientations) for the pixels within the cell. In each cell, HOG feature extraction computes centered horizontal and vertical gradients orientation and magnitudes with no smoothing. Finally, the histograms are normalised according to the histograms or nearby cells - block-. The combination of these histograms then represents the descriptor, such that the local object appearance and shape within an image is described by the distribution of intensity gradients or edge directions. The main steps are summarized in:

1) Compute gradients in the cell region to be described
2) Put them in bins according to orientation
3) Group the cells into large blocks
4) Normalize each block

B. GDCV Embedding

Once the HOG features (X_{HOG}) are calculated, our proposed system aims to find a linear mapping or projection onto a feature manifold where the correspondence between the input image and their angular pose is easier to be estimated than in the original space.

While many dimensionality reduction methods such as Principal Component Analysis (PCA) [27] or Linear Discriminant

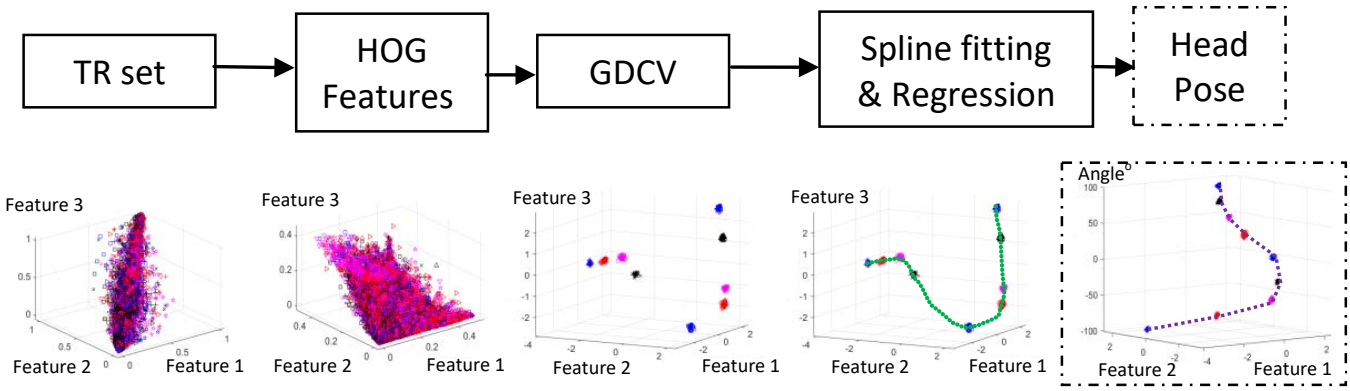

Fig. 1: Proposed head pose estimation pipeline and generated subspaces for each component during the training process.

Analysis (LDA) [28]can be applied to calculate this embedding (see Section III-C), two particular characteristics of the head pose estimation problem should be taken into consideration. First, poses corresponding to the same or very close angles should be kept together after the projection, while poses with very different angles should be separate as much as possible, in order to achieve an effective regression later. Second, the method should be able to cope with the large dimensionality of the face image data (and the even larger HOG feature dimensionality) in comparison with the available number of samples in the training set, which leads to the well-known Small Sample Size (SSS) problem [29] that produces singular matrices during computation.

Generalized Discriminative Common Vectors (GDCV) [30], [31] is proposed in our framework since it combines both properties. On the one hand, it provides discriminative subspaces and exhibits good generalization properties in a wider range of applications in computer vision and machine learning, regardless of the SSS assumption. On the other hand, GDCV is a supervised technique which makes use of the class information (in our case, the angle associated to the data sample) to obtain the most discriminative space by maximizing the distance between classes while minimizing the distance between the samples within the same class. In our setup, the classes are the possible angles for the yaw and the pitch. Although the angle estimation problem is a continuous problem in reality, which would produce an infinity number of classes, in practice, the number of angles in training is discrete and finite, since it is limited by the acquisition process and the number of steps between two poses in the training set.

GDCV method divides the feature space into the range and the null subspaces, being the later important for extracting useful discriminative features for the final regression. Thus, it generates a linear mapping onto the extend null space of its within-class scattered matrix in which all training of the same class collapse into the generalized common vectors, whose scatter is at the same time maximized. Formally, let the training set after HOG feature extraction X_{HOG} be composed of c classes, where every class j has m_j samples. The total number of samples in the calibration set is $M = \sum_{j=1}^{c} m_j$. Let x_j^i be a d-dimensional column vector of X_{HOG} which denotes the i^{th} sample from the j^{th} class. The within-class scatter matrix, S_w^X, is defined as,

$$S_w^X = \sum_{j=1}^{c} \sum_{i=1}^{m_j} (x_j^i - \bar{x}_j)(x_j^i - \bar{x}_j)^T = X_c X_c{}^T \tag{1}$$

where $\bar{x}_j$ is the average of the samples in the j^{th} class, and the centered data matrix, X_c consists of column vectors $(x_j^i - \bar{x}_j)$ for all $j = 1 \ldots c$ and $i = 1 \ldots m_j$.

The extension of the null space of S_w^X (which implies restricting the corresponding range space) is done from the eigendecomposition of S_w^X.

$$EVD(S_w^X) : U_r \Lambda_r U_r^T \tag{2}$$

where $U_r \in \mathbb{R}^{d \times r}$ are the eigenvectors associated to the nonzero eigenvalues Λ_r. The scattering added to the null space can be measured as the trace $tr(U_\alpha^T S_w^X U_\alpha)$. This quantity is zero when no directions are removed, $U_\alpha = U_r$, and increases as more and more important directions disappear from U_r. Consequently, the scattering preserved after a projection, U_α, can be written as follows

$$\alpha = 1 - \frac{tr(U_\alpha^T S_w^X U_\alpha)}{tr(S_w^X)}$$

The projection basis fulfilling the above conditions for a given value of α can be obtained through U_r, such that r is reassigned. The GDCV method is presented in Algorithm 1.

Any sample x_i can be projected in the discriminative subspace, for an easier classification, by using the projection matrix W_{GDCV}, according to

$$x_i^{gdcv} = W_{GDCV}^T \cdot (x_i - \bar{x}_{gcv}) \tag{3}$$

Given the usual bias of the training set to certain angles, since datasets are usually recorded at regular intervals, we exploit this feature in our advantage to reduce noise in the projection in those cases. Specifically, the previous projected sample i is refined to the location of the closest discriminative common vector j if this distance $d_{i,j}$ is below a small threshold

Algorithm 1:

GDCV method.

Input: $X \in \mathbb{R}^{d \times M}$, α.
Output: W_{GDCV}, $\bar{x}_{gcv}$.

1) Compute U_α such that $S_w^X = U_r \Lambda_r U_r^T$ where Λ_α contains the smallest eigenvalues in Λ_r and $tr(\Lambda_\alpha) = \alpha \cdot tr(\Lambda_r)$.
2) Project class means as $x_{gcv}^j = \bar{x}_j - U_\alpha U_\alpha^T \bar{x}_j$. These are the so-called generalized common vectors of each class.
3) Define $X^{com} = [x_{gcv}^1 \ldots x_{gcv}^c]$ and let X_c^{com} be its centered version with regard to the mean, $\bar{x}_{gcv} = \frac{1}{c}\sum_{j=1}^{c} x_{gcv}^j$.
4) Obtain the projection $W_{GDCV} \in \Re^{d \times (c-1)}$ such that $tr(W_{GDCV}^T X_c^{com} X_c^{comT} W_{GDCV})$ is maximum.

(see eq. 4). Otherwise, the projection remains unchanged as given by eq. 3.

$$x_i^{gdcv} = \begin{cases} x_{gcv}^j & if \quad d_{i,j} < \quad th_{gdcv_j} \\ x_i^{gdcv} & otherwise \end{cases} \tag{4}$$

where $d_{i,j}$ is the cosine distance:

$$d_{i,j} = \left(1 - \frac{x_{gcv}^j x_i^{gdcv^T}}{(x_{gcv}^j x_{gcv}^{j\ T})(x_i^{gdcv} x_i^{gdcv^T})}\right)$$

The threshold for each common vector is calculated as a third of the minimum distance to all other common vectors, that is:

$$th_{gdcv_j} = \min_{j'}(d_{j,j'})/3 \tag{5}$$

C. Multivariate Regression

After the manifold has been created, regression in such discriminative embedding space ($W_{GDCV}^T X_{HOG}$) is learned to generate the final pose estimation. This regression consists of two parts. First, a B-splines is used to construct a curve Y that has the best fit to the project samples, where the control points are the x_j^{gdcv}. This spline allows explicitly introducing the continuous and smooth transition between classes, inherent to the nature of the angular problem under consideration. Second, a multiple linear regression to estimate the relationships between the previous curve Y and the final angle(s) to be estimated Z is calculated.

B-splines: B-splines or Basis-splines [32] are mathematical curves with convenient properties. The curve reconstruction problem is to find a B-spline function f such that the geometric distance between the implicit curve $f(x_i, Y(x_i)) = 0$ and the point clouds be as small as possible. Meanwhile, the curve is expected to have a good quality, for which they use the condition that the implicit curve has a minimal simplified thin-plate energy. A curve $Y(x_i)$ is defined in terms of the P_k control points and the $B_k(x_i)$ B-spline basis functions.

$$Y(x_j^i) = \sum_{k=1}^{n} P_k B_k(x_j^i) \tag{6}$$

where the previously computed generalized discriminative common vectors are used as control points $P = [x_1^{gdcv}, \cdot, x_j^{gdcv}]$, and each basis function $B_k(x_i)$ is a piecewise polynomial with compact support determined by the position of the knots.

Multiple linear regression: The final prediction of the estimated head pose angles is provided by a regression model that describes the relationship between the dependent variables, $Z = [yaw, pitch]$, and one or more independent (explanatory) variables, Y in our case. In the particular case of multiple linear regression, the general model can be written as follows:

$$Z = \mathbf{Y}\beta + \varepsilon \tag{7}$$

which is equivalent to:

$$\begin{pmatrix} z_{11}.. & z_{1d'} \\ \vdots & \vdots \\ z_{M1}.. & z_{Md'} \end{pmatrix} = \begin{pmatrix} 1 & y_{11}.. & y_{1k} \\ \vdots & \vdots & \vdots \\ 1 & y_{M1}.. & y_{Mk} \end{pmatrix} \begin{pmatrix} \beta_{01}.. & \beta_{0d'} \\ \vdots & \vdots \\ \beta_{k1}.. & \beta_{kd'} \end{pmatrix} + \begin{pmatrix} \varepsilon_{11}.. & \varepsilon_{1d'} \\ \vdots & \vdots \\ \varepsilon_{M1}.. & \varepsilon_{Md'} \end{pmatrix}$$

and

$$z_{il} = \beta_{0l} + \beta_{1l} y_{i1} + \beta_{2l} y_{i2} + \ldots + \beta_{kl} y_{ik} + \varepsilon_{il}$$

is the ith response, $i = 1, \ldots, M$, of the lth output, $l = 1, \ldots, d'$. $\beta_{k'l}$ is the k'th regression coefficient, $k' = 1, \ldots, k$, and ε_{il} is the ith noise term, which models the random error. $d' = 2$ in our problem, since yaw and pitch are estimated, but the method can be tailored to only estimate one of them, or further extend to also estimate the roll angle. k is given by the dimensionality of the embedding space in the previous step, $k = (c-1)$.

Given a set of training data Y and their corresponding solution Z, the regression parameters can be easily estimated as:

$$\beta = (\mathbf{Y}^T \mathbf{Y})^{-1}\ \mathbf{Y}^T\ Z \tag{8}$$

D. Training and testing process

Figure 2 presents the main steps of the training framework propose as well as the learned parameters.

In the test process, new samples' head poses are calculated following the work flow shown in figure 3. Firstly, the HOG features for the testing sample x_{test} are computed, then this is projected into the discriminative subspace by using W_{GDCV}. The distances $d_{test,j}$ are calculated between the test sample and the generalized discriminative common vectors x_j^{gdcv}. If $d_{test,j} < th_{gdcv_j}$ the test sample is replaced by its corresponding x_j^{gdcv} as show 4. Finally, by projecting into the curve Y, the angle estimation of the test sample is predicted using the multiple linear regression β.

In summary, let the training set X be composed of M samples and their corresponding angles Z, Algorithm 2 shows the main steps of our framework propose.

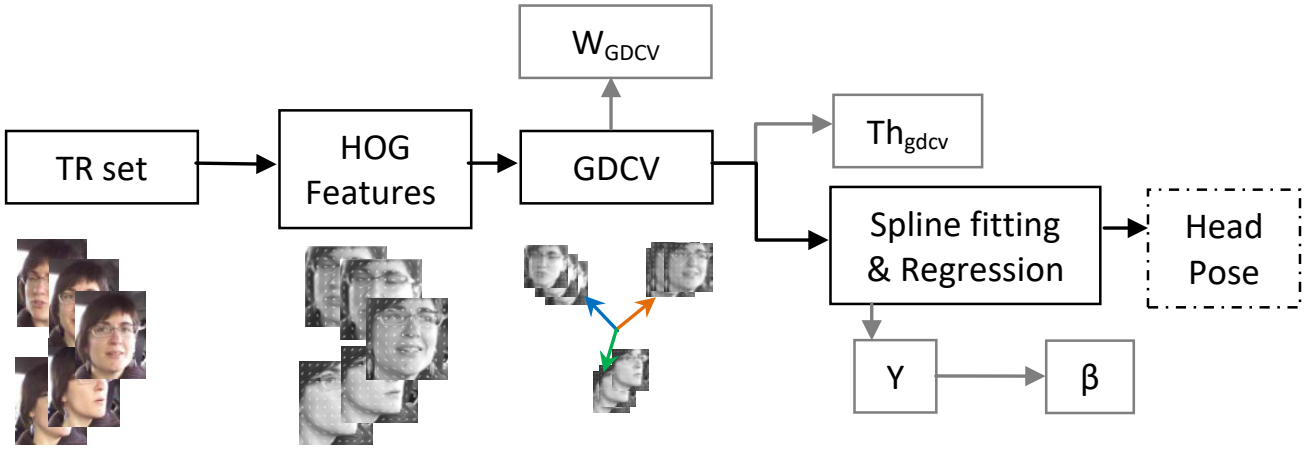

Fig. 2: Training methodology and the resulting learned parameters (in light gray).

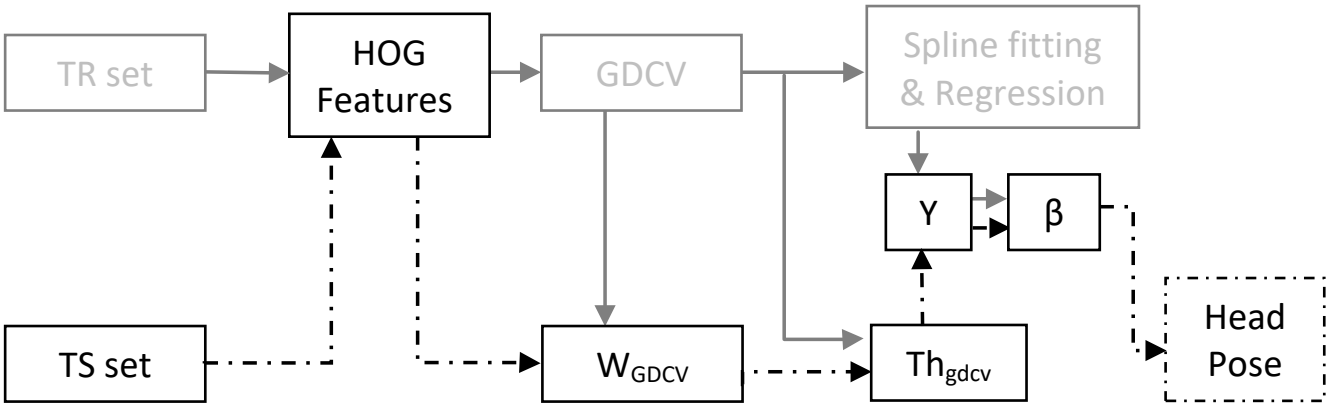

Fig. 3: Test methodology.

Algorithm 2:

Main steps of the framework propose to head pose estimation.

Input: $X \in \mathbb{R}^{d \times M}$, $Z \in \mathbb{R}^{M \times 1or2}$
Output: W_{GDCV}, x^j_{gcv}, th_{gdcv_j}, Y, β
Training:

1) Compute $X_{HOG} = HOG(X)$.
2) Compute $GDCV(X_{HOG})$ and obtain W_{GDCV}, x^j_{gcv}, and th_{gdcv_j}.
3) Build Y by using Eq. 6, where $P_k = x^j_{gcv}$ and $k = j$.
4) Compute β by using Eq. 8.

Test: x_{test}

1) Compute $x_{test_{HOG}} = HOG(x_{test})$.
2) Project $x_{test_{HOG}}$ as $x_{test_{GDCV}} = W^T_{GDCV} x_{test_{HOG}}$.
3) Calculate the distance $d_{test,j}$ between $x_{test_{GDCV}}$ and the x^j_{gcv}. If $d_{test,j} < th_{gdcv_j}$, $x_{test_{GDCV}} = x^{gdcv}_j$.
4) Project $x_{test_{GDCV}}$ into the curve Y.
5) The angular prediction is allocated as $x_{test_{GDCV}} \beta$.

III. Experiments and Results

In order to ensure an exhaustive evaluation, our method is validated with six publicly available standard datasets, CMU-PIE [33], Taiwan [34], PRIMA [35] CASPEAL-1 and 2 [36] and DrivFace [7]. This selection was chosen to ensure most possible situations and poses are considered. Thus, maximum angular deviations (from -90 to 90 degrees) for both yaw are present in CMU-PIE and Taiwan and for both yaw and pitch in PRIMA. High angular resolution (small steps in angles) were used in Taiwan. Datasets with high (CASPEAL-1) and low (CASPEAL-2) resolution images are also included, as well as datasets with few (CMU-PIE) and many (CASPEAL-2)

images and users. Finally, the DrivFace [7] dataset is used for testing under real conditions the different derived models. The table in Figure 4 shows the main characteristics of each datasets, where the size of face crops has been normalized to 80×80.

Our approach is compared against three state-of-the-art head-pose estimation methods. The first one, called HPE, proposed by Drouard et al. [9], [10], is a manifold-embedding approach similar to us in the use of linear regressions and HOG features, which can be trained in our exact experimental setup due to the availability of the code [1]. Both other methods, the Discriminative Response Map Fitting (DRMF) method [11] and OPENFACE [12] are pretrained model approaches. DRMF, based on facial landmarks, uses discriminative regression with constrained local models to reconstruct unseen response maps[2]. OpenFace is a CNN-based framework capable of facial landmark detection, head pose estimation, facial action unit recognition and eye-gaze estimation.[3].

In our validation, two main scenarios are considered. In the first scenario, an intra-set experimental setup is considered where a model is generated for every dataset considered and both training and testing partitions for every model are coming from the same dataset. A second and more challenging cross-dataset scenario in also considered, where a model is training in all datasets except the one used in testing (all against one) to simulate more realistic conditions.

The parameters of our method used were set to conventional values according to their authors and without any particular optimization. Once chosen, they were kept constant for all experiments. Thus, to obtain the HOG features, the gradient computation uses a central difference filter [-1 0 1] and using forward difference at the image borders, the gradient directions are between -180 and 180 degrees measured counter clockwise from the positive x axis, with 9 bins. The size of a HOG cell in pixels is 5x5, the block size is 2x2. [4]. Regarding the GDCV method, the α value was set to 0.95 as in [37], [38], and the cosine distance is used to calculated the th_{gdcv_j}. Finally, the Boor's algorithm [39] is used for generating the spline curves [5]. All algorithms have been run on a computer with a Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz, 3601 Mhz, and 32-GB RAM.

A. First scenario: Intra-dataset validation

In this experiment, the method is trained and tested in different partitions of the same dataset. Specifically, all five standard datasets are used, and a model for each dataset under consideration is generated (training) with the 50% of the samples and tested with the remain 50%, Cross validation is applied as evaluation protocol to avoid bias to a particular training/testing split, where each experiment is run 10 times with different random training/testing sample choices. Our approach is compared against HPE in the same training/testing setup. DRMF and OPENFACE are also added to the comparison for reference according to their results in the testing partitions, but using the best trained model provided by the authors.

Experimental results are provided for both cases when all the head image (and surrounded background) is used as input by the system for or when only the face crop is provided (with the exception of OPENFACE whose software requires the full head to fit their 3D model). Table I shows the average result over the iterations as well as dispersion for the full head image variation, and Table II the same results when using the face crop only.

Dataset	DRMF		OPENFACE	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	17.8 ± 7.2	–	39.3 ± 13.6	–
Taiwan	33.3 ± 14.0	–	44.3 ± 18.9	–
PRIMA	53.3 ± 18.6	45.2 ± 5.2	54.5 ± 20.9	44.8 ± 13.0
CASPEAL-1	28.4 ± 16.3	26.6 ± 5.0	27.3 ± 16.2	26.1 ± 13.3
CASPEAL-2	9.9 ± 8.8	27.8 ± 4.3	15.6 ± 7.6	24.2 ± 6.5
	Our		HPE	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	1.2 ± 3.9	–	18.5 ± 5.7	–
Taiwan	13.1 ± 13.8	–	37.6 ± 31.9	–
PRIMA	17.1 ± 14.6	29.9 ± 23.6	33.2 ± 33.9	47.0 ± 33.7
CASPEAL-1	1.6 ± 5.9	10.1 ± 11.8	35.0 ± 25.7	28.5 ± 7.6
CASPEAL-2	3.9 ± 3.5	13.9 ± 11.7	13.7 ± 4.1	37.2 ± 11.7

TABLE I: Average error in degrees for the intra-dataset experiment using the full head image as input.

Several relevant conclusions can be achieved from these results. First, our proposed approach provides the best results in all cases for both possible inputs, improving greatly the next best result, HPE. As it could be expected, since less distracters are present in the image, all methods behave significantly better when using only the face crop (except in a couple of cases whose difference is not significant). However, this assumes that a face detector/segmentation algorithm is available with almost perfect performance, which is challenging and unrealistic in real-life scenarios. In these situation, it may be easier to provide the full head image. While HPE increases the error between 0 and 37 degrees for the yaw and up to 25 degrees for the pitch, depending on the dataset, our approach only increases between 0 and 9 degrees for the yaw and up to 20 degrees for the pitch. It can also be noticed that pitch angle seems more difficult to be estimated, although it is likely that this is the result of having less training examples since not all datasets have images with varying pitch. An unusually large error value of the pitch in the CASPEAL-2 is given by our method, which is caused by the limited information contained on low resolution face crops.

Both DRMF and OPENFACE methods provide very poor result, some of which can be considered almost random, since they have not been trained on the testing image types. This is therefore a not fair or conclusive comparison but they have been added here to illustrate the difficulty of generate useful models in real life applications, as well as the limitations of current methods. Next subsection makes emphasis in this problem and fair comparison.

[1]The HPE code is available at https://team.inria.fr/perception/research/head-pose/. In our experimentation K = 5.

[2]The DRMF code is available at https://ibug.doc.ic.ac.uk/resources/drmf-matlab-code-cvpr-2013/

[3]The OPENFACE code is available at https://www.cl.cam.ac.uk/research/rainbow/projects/openface/

[4]The extractHOGFeatures function of Matlab is used.

[5]The code is available at http://www.mathworks.com/matlabcentral/fileexchange/27374-b-splines

Dataset	Angle	c	m_j	Subjects	Yaw	Pitch	Image	Face Crop
1. CMU-PIE [33]	-90:22.5:90	9	67	67	x		120×100	80×80
2. Taiwan [34]	-90:5:90	37	180	90	x		120×160	80×80
3. PRIMA [35]	-90:15:90	13	209	15	x		288×384	80×80
	-90:15:90	9	–			x		
4. CASPEAL-1 [36]	-67: 22:67	7	303	101	x			80×80
	-30: 30:30	3	707			x	480×360	80×80
5. CASPEAL-2 [36]	-45: 15:45	7	2815	939	x		80×80	40×40
	-30: 30:30	3	–			x		80×80
6. DrivFace [7]	-45:15:45	7	–	4	x		640×480	80×80

Fig. 4: Datasets used in validation along with their corresponding details. c is the number of classes. m_j is the number of samples per class.

Dataset	Our		HPE		DRMF	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	1.9 ± 4.8	–	18.1 ± 5.3	–	16.1 ± 8.2	–
Taiwan	5.8 ± 4.6	–	6.9 ± 5.0	–	30.7 ± 17.6	–
PRIMA	8.1 ± 7.8	9.6 ± 10.0	13 ± 9.0	22.6 ± 12	42.0 ± 17.4	46.0 ± 6.5
CASPEAL-1	1.0 ± 3.6	2.3 ± 7.2	20.4 ± 12.7	30.4 ± 5.5	24.0 ± 15.7	25.1 ± 5.7
CASPEAL-2	2.9 ± 4.1	30.6 ± 18.8	12.4 ± 10.7	30.5 ± 6.5	13.4 ± 7.5	26.6± 4.2

TABLE II: Average error in degrees for the intra-dataset experiment using the face crop as input.

B. Second scenario: Cross-dataset validation

In order to compare all methods under equal conditions, as well as present a more challenging and realistic scenarios, we perform a cross dataset experiment. Specifically, an all-against-one strategy is adopted for our method and HPE, where a model using all images in four standard datasets except one are used for training and the fifth dataset is used for testing. This is repeated generating a different model for all possible combinations. This experiment also aims to validate the previous conclusions and results and ensure that the validity of our approach is not the result of overfitting. The pretrained approaches DRMF and OPENFACE are added in the comparison using the same test sets but this time the comparison is fair (even if the training sets are different) since no methods have seen the testing type of images.

Table III and IV shows the average result over the iterations as well as dispersion for the cases that the full image or the face crops are used, respectively. It can be notice how all reported errors for HPE and our method increases due to the most challenging problem, getting closer results to the pretrained methods. For the full image experiment, our method still reports the best results in all cases. For the face crop experiment, results are not so clear and HPE provides in many cases similar or better results. However, our approach is still providing the best pitch estimation without having a significantly lower yaw estimation, and without restrictions regarding the training data capture, such as the number of classes or the angular resolution.

Finally, in order to provide our best possible system for its application in real scenario, a final experiment is designed where the model is trained using Taiwan and PRIMA datasets. This is due to them having the best pitch and yaw resolution. Face crops is used as input due to its superior performance demonstrated in previous experiments. This model is tested in all remaining datasets, including the DrivFace [7] dataset which contains real variations such as illumination changes, vibrations and imperfect face crops, and compared against all

Dataset	Our		HPE		OPENFACE	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	30.8 ± 21.7	–	N/A	–	39.4 ± 9.4	–
Taiwan	33.3 ± 23.5	–	N/A	–	37.2 ± 16.7	–
PRIMA	39.9 ± 25.1	41.6 ± 14.3	55.5 ± 44.2	N/A	32.1 ± 17.4	67.6 ± 35.2
CASPEAL-1	27.3 ± 22.8	16.9 ± 14.4	27.7 ± 22.3	37.5 ± 11.7	36.8 ± 15.6	29.3 ± 17.6
CASPEAL-2	13.8 ± 14.8	16.3 ± 15.6	27.4 ± 19.1	35.5 ± 10.6	18.3 ± 9.7	23.2 ± 7.9

TABLE III: Average error in degrees for the cross-dataset experiment using the full head image as input.

Dataset	Our		HPE		DRMF	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	26.9 ± 17.5	–	N/A	–	33.6 ± 15.5	–
Taiwan	18.0 ± 13.8	–	N/A	–	30.1 ± 16.7	–
PRIMA	21.5 ± 17.0	32.9 ± 11.6	13.7 ± 10.3	N/A	40.5 ± 18.0	46.4 ± 7.0
CASPEAL-1	13.2 ± 11.6	10.9 ± 11.3	10.4 ± 10.9	34.1 ± 9.3	24.4 ± 15.7	25.4 ± 5.8
CASPEAL-2	9.7 ± 9.8	7.4 ± 12.4	10.1 ± 12.8	30.5 ± 4.2	13.2 ± 8.4	26.9 ± 4.3

TABLE IV: Average error in degrees for the cross-dataset experiment using the face crop as input. N/A is reported for HPE in some cases due to the limitation of the algorithm to run, such as a lower number of classes in training than in testing.

Dataset	DRMF		OPENFACE	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	33.6 ± 15.5	–	39.4 ± 9.4	–
CASPEAL-1	24.4 ± 15.7	25.4± 5.8	36.8 ± 15.6	29.3 ± 17.6
CASPEAL-2	13.2 ± 8.4	26.9 ± 4.3	18.3 ± 9.7	23.2 ± 7.9
DrivFace	18.5 ± 9.8	–	16.0 ± 11.1	–
	Our		HPE	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	16.8 ± 17.6	–	10.6 ± 9.1	–
CASPEAL-1	11.2 ± 10.7	9.8 ± 11.6	13.2 ± 10.4	14.0 ± 8.9
CASPEAL-2	9.4 ± 9.0	10.3 ± 11.8	12.5 ± 16.1	14.4 ± 8.9
DrivFace	16.1 ± 14.8	–	19.5 ± 8.2	–

TABLE V: State of the art comparison, by using Taiwan and PRIMA datasets in the training.

other competitors (HPE with the same training and DRMF using the best training provided by the authors).

Table V shows the comparative among all methods. It can be seen how our approach with a carefully selected training provides the best performance in almost all cases (expect for CMU-PIE, the smallest set where HPE gives the best result), with errors ranging between 9 and 16 degrees for the yaw and around 10 degrees for the pitch, which are acceptable for most applications.

C. Ablation Studies

In order to justify and validate our pipeline, we repeat the previous cross-dataset experiment in Table 5 but removing or replacing with conventional approaches some of the modules in our pipeline.

First, HoG feature extraction is removed and image raw pixels are given to GDCV as direct input. The comparison is shown in Table VI. It can be observed how the use of HOG features help to obtain a more discriminative subspace and a better performance, reducing the angular error between 12 and 31 degrees, depending the dataset. This is particularly noticeable in realistic conditions (DrivFace), where illumination changes are frequent and can affect greatly the raw pixel values.

In a second ablation experiment, the discriminant subspace is generated using the well-known Linear Discriminant Analysis (LDA) [40] instead of our proposed GDCV. Results in Table VII indicate that GDCV is a technique better suited for the head pose estimation problem, able to produce a more discriminative embedding space.

Dataset	With HOG		Without HOG	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	16.8 ± 17.6	–	28.7 ± 25.4	–
CASPEAL-1	11.2 ± 10.7	9.8 ± 11.6	28.6 ± 28.4	19.8 ± 13.0
CASPEAL-2	9.4 ± 9.0	10.3 ± 11.8	22.7 ± 22.8	21.5 ± 12.7
DrivFace	16.1 ± 14.8	–	47.1 ± 30.8	–

TABLE VI: Average error in degrees with and without HOG feature extraction.

Dataset	GDCV		LDA	
	Yaw^o	Pitch^o	Yaw^o	Pitch^o
CMU-PIE	16.8 ± 17.6	–	26.5 ± 34.4	–
CASPEAL-1	11.2 ± 10.7	9.8 ± 11.6	19.9 ± 24.8	19.6 ± 8.5
CASPEAL-2	9.4 ± 9.0	10.3 ± 11.8	15.9 ± 22.1	20.1 ± 8.7
DrivFace	16.1 ± 14.8	–	42.3 ± 35.5	–

TABLE VII: Average error in degrees using LDA or GDCV as discriminant subspace.

IV. Conclusions

In this paper, we propose a novel appearance-based head estimation system for both yaw and pitch estimation. Our system combines HOG feature extraction with an GDCV manifold embedding, that takes the granular high dimensional nature of the problem, and multivariate regression, that considers the continuous and smooth continuity of the estimated angles by applying splines. Our system demonstrates flexibility to work with raw head images, wildly available in real conditions, or more refined facial crops, assuming a good face detector is available. Our approach achieves state-of-art performance in an exhaustive experimental validation comprising six different datasets and both intra-set and cross-dataset experiments. The final performance surpasses the other three methods in the comparison, including CNN-based methods, with angular errors between 9 and 17 degrees, and was evaluated in a realistic datasets for autonomous driving.

References

[1] Y. Tian, L. Brown, J. Connell, S. Pankanti, A. Hampapur, A. Senior, and R. Bolle, "Absolute head pose estimation from overhead wide-angle cameras," in *AMFG*, 2003, pp. 92–99.

[2] M. Goodrich and A. Schultz, "Humanrobot interaction: A survey," *Foundations and Trends in HumanComputer Interaction*, vol. 1, no. 3, pp. 203–275, 2008.

[3] R. Stiefelhagen, "Tracking focus of attention in meetings," in *ICMI*, 2002, pp. 273–280.

[4] G. C. Lee, C. K. Loo, and L. Chockalingam, "An integrated approach for head gesture based interface," *Applied Soft Computing*, vol. 12, no. 3, pp. 1101–1114, 2012.

[5] B.-G. Lee and W.-Y. Chung, "Driver alertness monitoring using fusion of facial features and bio-signals," *Sensors*, vol. 12, no. 7, pp. 2416–2422, 2012.

[6] D. Mikio, K. Matti, and K. JuhaM., "Measurement of driver visual attention capabilities using real-time ufov method," *International Journal of Intelligent Transportation Systems Research*, vol. 9, no. 3, pp. 115–127, 2011.

[7] K. Diaz-Chito, A. Hernández-Sabaté, and A. López, "A reduced feature set for driver head pose estimation," *Appl. Soft Comput.*, vol. 45, pp. 98–107, 2016.

[8] N. Alioua, A. Amine, A. Rogozan, A. Bensrhair, and M. Rziza, "Driver head pose estimation using efficient descriptor fusion," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, 2016.

[9] V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, and R. Horaud, "Head Pose Estimation via Probabilistic High-Dimensional Regression," in *IEEE International Conference on Image Processing*, ser. Proceedings of the IEEE International Conference on Image Processing, no. 4624 – 4628, 2015.

[10] V. Drouard, R. Horaud, A. Deleforge, S. Ba, and G. Evangelidis, "Robust head-pose estimation based on partially-latent mixture of linear regressions," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1428 – 1440, 2017.

[11] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3444–3451.

[12] T. Baltruaitis, P. Robinson, and L. P. Morency, "Openface: An open source facial behavior analysis toolkit," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–10.

[13] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: a survey," *IEEE T-PAMI*, vol. 31, no. 4, pp. 607–626, 2009.

[14] A. Narayanan, R. M. Kaimal, and K. Bijlani, "Estimation of driver head yaw angle using a generic geometric model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3446–3460, 2016.

[15] C. Wang, Y. Guo, and X. Song, "Head pose estimation via manifold learning," in *Manifolds - Current Research Areas*. InTech, 2017, ch. 06.

[16] J. Chen, J. Wu, K. Richter, J. Konrad, and P. Ishwar, "Estimating head pose orientation using extremely low resolution images," in *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, 2016, pp. 65–68.

[17] D. Huang, M. Storer, F. D. la Torre, and H. Bischof, "Supervised local subspace learning for continuous head pose estimation," in *CVPR 2011*, 2011, pp. 2921–2928.

[18] M. A. Haj, J. Gonzlez, and L. S. Davis, "On partial least squares in head pose estimation: How to simultaneously deal with misalignment," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2602–2609.

[19] C. Wang and X. Song, "Robust head pose estimation using supervised manifold projection," in *IEEE International Conference on Image Processing*, 2012, pp. 161–164.

[20] ——, "Robust head pose estimation via supervised manifold learning," *Neural Networks*, vol. 53, no. Supplement C, pp. 15 – 25, 2014.

[21] X. Peng, J. Huang, O. Hu, S. Zhang, A. Elgammal, and D. Metaxas, "From circle to 3-sphere: Head pose estimation by instance parameterization," *Comput. Vis. Image Underst.*, vol. 136, no. C, pp. 92–102, 2015.

[22] J. Foytik and V. A. Asari, "A two-layer framework for piecewise linear manifold-based head pose estimation," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 270–287, 2013.

[23] B. Ahn, P. Jaesik, and K. I. So, "Real-time head orientation from a monocular camera using deep neural network," in *Asian Conference on Computer Vision*, 2015, pp. 82–96.

[24] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods," *Pattern Recognition*, vol. 71, no. Supplement C, pp. 132 – 143, 2017.

[25] N. McLaughlin, J. M. del Rincon, and P. Miller, "Data-augmentation for reducing dataset bias in person re-identification," in *Proceedings of 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2015.

[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[27] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press, 1990.

[28] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[29] L. Chen, H. Hong-Yuan, M. Ko, J. Lin, and G. Yu, "A new lda-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.

[30] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "Discriminative common vectors for face recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 4–13, 2005.

[31] A. Tamura and Q. Zhao, "Rough common vector: A new approach to face recognition," in *IEEE Intl. Conf. on Syst, Man and Cybernetics,*, 2007, pp. 2366–2371.

[32] T. Hastie, R. R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2009.

[33] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition*, 2002.

[34] "Taiwan face dataset," http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm#Our_Database_, accessed: 2017-09-14.

[35] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *FG Net Workshop on Visual Observation of Deictic Gestures*, vol. 6, 2004.

[36] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The cas-peal large-scale chinese face database and baseline evaluations," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 1, pp. 149–161, 2008.

[37] K. Diaz-Chito, F. Ferri, and W. Diaz-Villanueva, "Incremental generalized discriminative common vectors for image classification," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 8, pp. 1761–1775, 2015.

[38] K. Diaz-Chito, F. Ferri, and A. Hernández-Sabaté, "An overview of incremental feature extraction methods based on linear subspaces," *Knowl.-Based Syst.*, vol. 145, pp. 219–235, 2018.

[39] C. D. Boor, *A practical guide to splines; rev. ed.*, ser. Applied mathematical sciences. Berlin: Springer, 2001.

[40] K. Fukunaga, *Introduction to Statistical Pattern Recognition (2Nd Ed.)*. Academic Press Professional, Inc., 1990.